

In the Claims:

1. (Currently Amended) A ~~generation~~ debugger for visual debugging of a ~~group of constraints~~ declarative language encapsulated constraint system, during a test generation process by a generator, comprising:

a generatorcollector for collecting generatinggeneration events during a test process comprising generation of objects and generation decisions for debugging said declarative language encapsulated constraint system, and

a systematic, graphical representation for relating respective generation objects and said generation decisions, during said test process. ~~declarative language encapsulated constraint system generation process by a generator.~~

2. (Original) The debugger of claim 1, wherein said graphical representation is as a two dimensional chart.

3. (Original) The debugger of claim 2, wherein said two-dimensional chart is based on generation events collected during the generation process and static analysis phase, each event reflecting a generation operation, and wherein generation entities for generating said generation events are presented on a first dimension of said chart and a second dimension represents an execution sequence, with generation events being displayed as aligned with their related generation entities.

4. (Original) The debugger of claim 1, further comprising: a data browser for interactive selection of generation entities to be viewed.

5. (Original) The debugger of claim 1, further comprising: a step tree for presenting a sequence of steps performed by the generator, for identifying the step where the computation diverged from the expected behavior.

6. (Original) The debugger of claim 1, further comprising: an event browser for displaying generation events.

7. (Original) The debugger of claim 1, further comprising: an order browser for displaying generation field order decisions.

8. (Currently Amended) A method for visual debugging of a constraint system, said constraint system being encapsulated in a declarative language, group of ~~constraints during a test generation process by a generator~~, comprising:

displaying a plurality of generation events collected during ~~the a~~ generation process of said constraint system by a generator, such that a relationship between said plurality of generation events and a plurality of generation entities for generating said generation events is graphically displayed, and wherein an order of execution of said generation entities is also graphically displayed, for visual debugging of the group of constraints.

9. (Original) The method of claim 8, further comprising:

viewing a plurality of generation events sequentially from a selected event.

10. (Original) The method of claim 9, wherein said sequence is displayed forward from said selected event.

11. (Original) The method of claim 9, wherein said sequence is displayed backward from said selected event.

12. (Currently Amended) A method for debugging a generation process with a user, comprising:

analyzing ~~the~~ a generation process of a constraint system, said constraint system being encapsulated in a declarative language, to extract a sequence of events from the generation process; and

displaying at least a portion of said sequence of events to the user in a visual display.

13. (Original) The method of claim 12, wherein said visual display includes a representation of at least one generated field from at least one event.

14. (Original) The method of claim 12, wherein said visual display includes a representation of at least one constraint from at least one event.

15. (Original) The method of claim 12, wherein said visual display includes a representation of at least one generation event related to a generation entity.

16. (Original) The method of claim 12, wherein said visual display includes at least one type of information displayed as a result of a selection by the user.

17. (Currently Amended) A ~~generation~~-debugger for debugging a generation process, comprising:

an analyzer for analyzing ~~the~~ a generation process of a constraint system, said constraint system being encapsulated in a declarative language, to extract a sequence of events from the generation process; and

a visual display for displaying information related to at least one of a field, a constraint, a generation event, a path of a generation event, and a combination thereof.

18. (Original) The generation debugger of claim 17, wherein said visual display further displays information related to an event collected during static analysis.

19. (Original) The generation debugger of claim 17, wherein said visual display further displays information related to an event collected during program execution.

20. (Original) The generation debugger of claim 17, wherein said information is represented with at least one icon and wherein said visual display further displays information when said icon is selected.

21. (Original) The generation debugger of claim 17, wherein said visual display further displays ordering information for a plurality of fields.

22. (Original) The generation debugger of claim 21, wherein said visual display further displays ordering information based on static analysis.

23. (Original) The generation debugger of claim 21, wherein said visual display further displays ordering information based on order computed dynamically

24. (Original) The generation debugger of claim 21, wherein said visual display further displays ordering information related to a group of fields selected through said visual display.